LLNL-PRES-558397

Lawrence Livermore National Laboratory

16th Annual Center for Advanced Signal and Imaging Sciences (CASIS)
Signal and Imaging Sciences Workshop, May 23, 2012

Statistical Feature Selection for non-Gaussian distributed Target Classes

May 23, 2012

Matthew J. Wilder, ENS, USN, Former MSEE Student, NPS
Grace A. Clark, Ph.D., IEEE Fellow
Eng/CED, Visiting Research Professor, NPS

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551
This work performed under the auspices of the U.S. Department of Energy by
Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

Disclaimer and Auspices Statements

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory in part under Contract W-7405-Eng-48 and in part under Contract DE-AC52-07NA27344.

Contact Information for Grace Clark

Research Engineer
Systems and Intelligence Analysis Section
Engineering Directorate
Lawrence Livermore National Laboratory
Contractor to the U.S. Dept. of Energy
7000 East Ave., L-130
Livermore, CA 94550
(925) 423-9759 (Office)
(925) 422-2495 (FAX)
clark9@llnl.gov

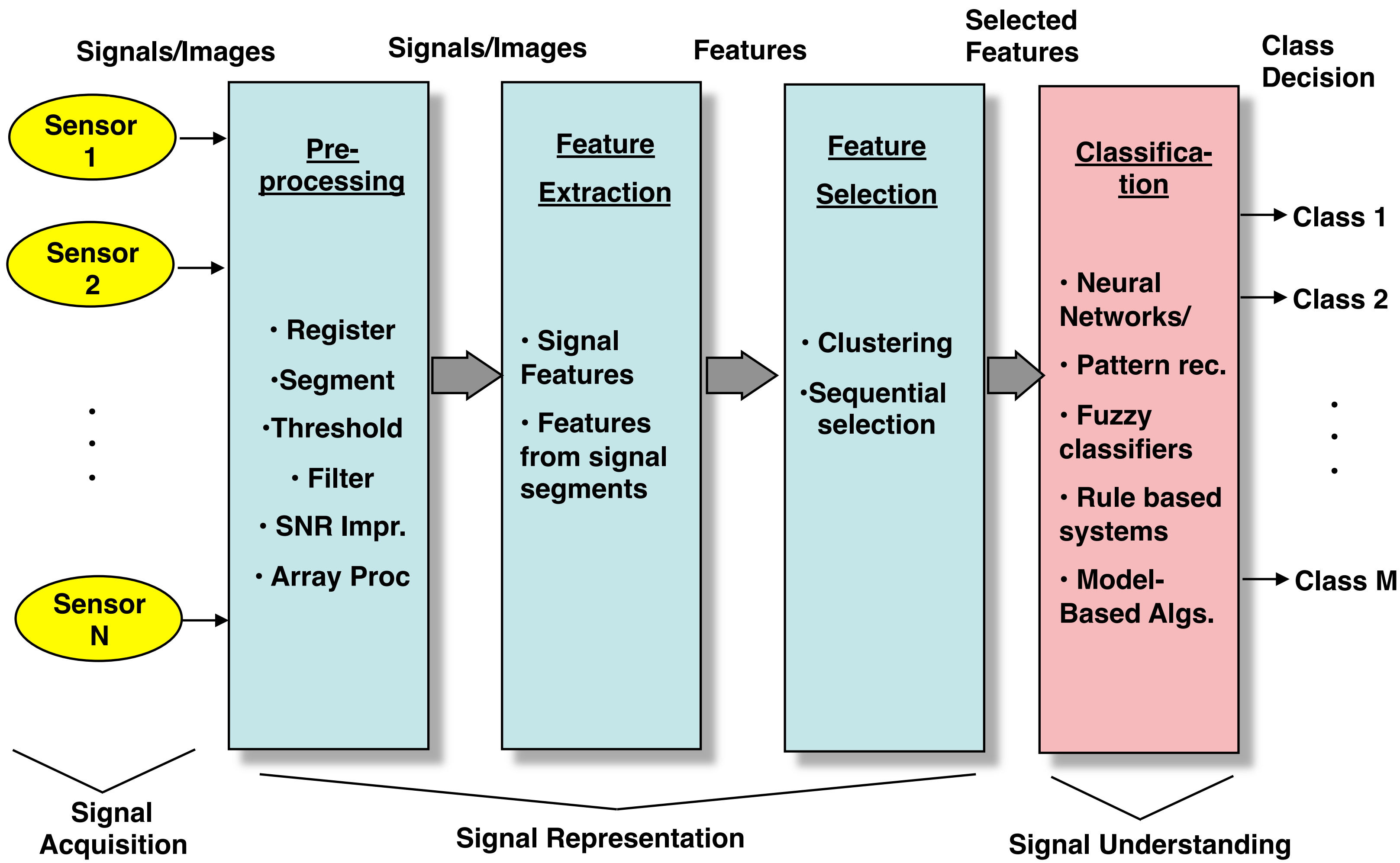
Grace Clark, Ph.D., IEEE Fellow, serves as Visiting Research Professor in the Center for Cyber Warfare at the Naval Postgraduate School (NPS), Monterey, CA, on Professional Research and Teaching Leave from the Lawrence Livermore National Laboratory, Livermore, CA. She earned BSEE and MSEE degrees from the Purdue University EE Honors Program and the Ph.D. ECE degree from the U. of California Santa Barbara. Her technical expertise is in statistical signal/image processing, estimation/detection, pattern recognition/machine learning, sensor fusion, communication and control. Dr. Clark has contributed more than 175 publications in the areas of acoustics, electro-magnetics and particle physics. Dr. Clark is a member of the ASA Technical Council on Signal Processing in Acoustics, as well as IEEE, SEG (Society of Exploration Geophysicists), Eta Kappa Nu and Sigma Xi.

The World of Acoustics Before Signal Processing

Agenda

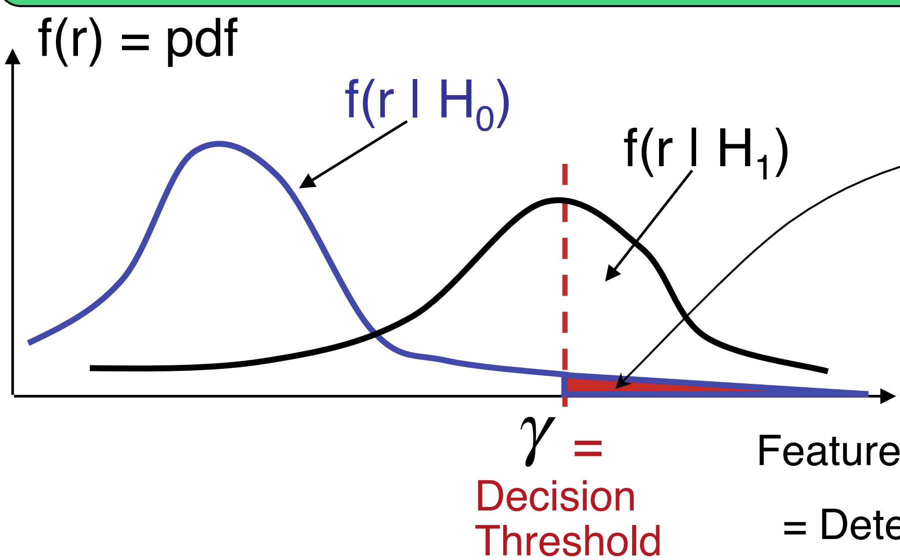
- Introduction
  - The Automatic Target Recognition Problem
  - Feature Selection Fundamentals
- Feature Selection for Gaussian Target Classes
  - Distance Measures
  - Subset Selection Algorithms
- Information-Theoretic Distance Measures
  - Divergence
  - Hellinger Divergence
- Density Estimation
- New Feature Selection Algorithm for Non-Gaussian Target Classes
- Experimental Results
- Discussion

Automatic Target Recognition Depends Heavily on the Judicious Choice of Signal / Image Features



The ROC Curve is Computed by Integrating Under the Conditional Probability Density Functions for a Given Threshold γ

r = Detection Statistic (e.g. Grey Scale Values)
For Example: Posterior Probabilities $P(H_1 \mid \underline{X})$ or $P(H_0 \mid \underline{X})$



$$P(H_1 \mid H_0) = P_{FA}(r_0) = \int_{\gamma}^{\infty} f(r \mid H_0)\, dr = 1 - P_{SPEC}(r_0)$$

$$P(H_1 \mid H_1) = P_D(r_0) = \int_{\gamma}^{\infty} f(r \mid H_1)\, dr = 1 - P(H_0 \mid H_0) = 1 - P_{MISS}(r_0)$$

$$P(H_0 \mid H_1) = P_{MISS}(r_0) = \int_{-\infty}^{\gamma} f(r \mid H_1)\, dr = 1 - P(H_1 \mid H_1) = 1 - P_D(r_0)$$

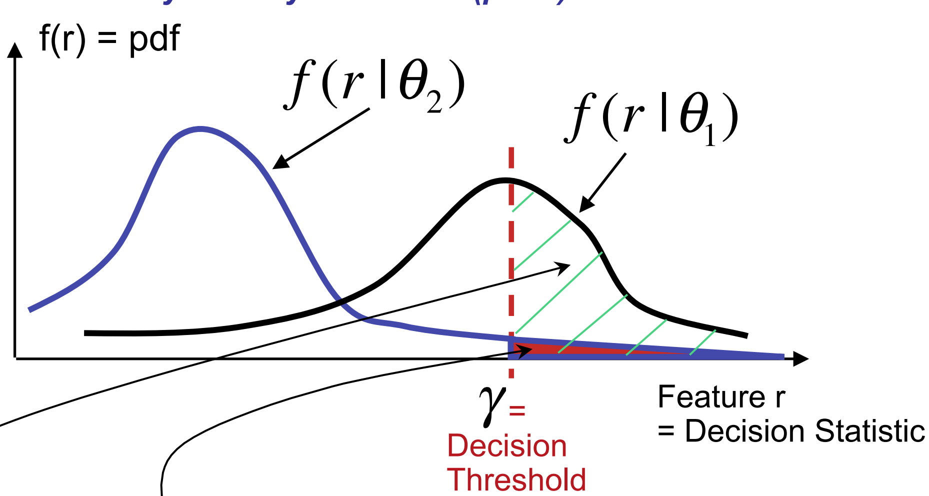
$$P(H_0 \mid H_0) = P_{SPEC}(r_0) = \int_{-\infty}^{\gamma} f(r \mid H_0)\, dr$$

Hypothesis Testing Generates a Receiver Operating Characteristic (ROC) Curve

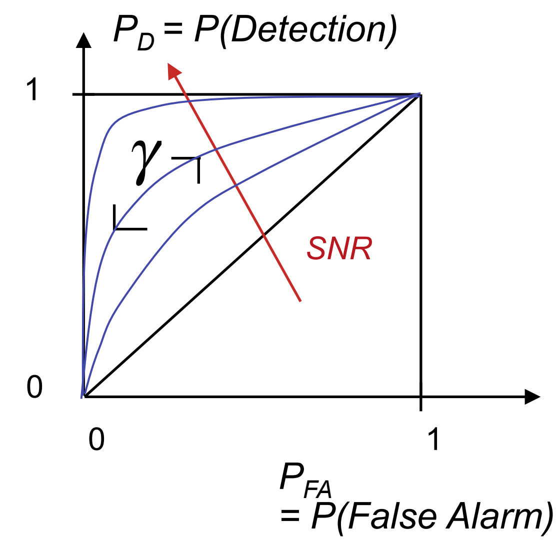
t = Time
x(t) = Signal of Interest
v(t) = Noise or "Background"
r(t) = x(t) + v(t) = Measurement
γ = Decision Threshold

$$\text{Hypothesis} \quad H_1\ (\text{Event}): \quad r(t) = x(t) + v(t)$$
$$\text{Hypothesis} \quad H_0\ (\text{Backgrond}): \quad r(t) = v(t)$$

Probability Density Functions (pdf's)



ROC Curve



$$P(False\ Alarm) = P_{FA}(\gamma) = \int_{\gamma}^{\infty} f(r \mid \theta_2)\, dr$$

$$P(Detection) = P_D(\gamma) = \int_{\gamma}^{\infty} f(r \mid \theta_1)\, dr$$

The Confusion Matrix is Used to Measure Classification Performance

Decision Rule: $\dfrac{f(\underline{X} \mid H_1)}{f(\underline{X} \mid H_0)} \geq \eta$

"Confusion Matrix" or Contingency Table for Binary Hypothesis Testing

Truth / Declaration	True Hypothesis H_0 (Null)	True Hypothesis H_1
Declared Hypothesis H_0(Null)	$P(H_0 \mid H_0) = P_{Spec} = Specificity$ $= \dfrac{\#\, H_0\ Samples\ \ Declared\ H_0}{\#\, H_0\ Samples}$	$P(H_0 \mid H_1) = P_{Miss} = P(Miss)$ $= \dfrac{\#\, H_1\ Samples\ \ Declared\ H_0}{\#\, H_1\ Samples}$
Declared Hypothesis H_1	$P(H_1 \mid H_0) = P_{FA} = P(False\ Alarm)$ $= \dfrac{\#\, H_0\ Samples\ \ Declared\ H_1}{\#\, H_0\ Samples}$	$P(H_1 \mid H_1) = P_D = P(Detection)$ $= \dfrac{\#\, H_1\ Samples\ \ Declared\ H_1}{\#\, H_1\ Samples}$

$$P(H_0 \mid H_0) + P(H_1 \mid H_0) = 1 \qquad P(H_0 \mid H_1) + P(H_1 \mid H_1) = 1$$

$$P_{cc} = P(Correct\ Classification) = P(H_0 \mid H_0)P(H_0) + P(H_1 \mid H_1)P(H_1)$$

$$P_e = P(Error) = 1 - P_{cc} = P(H_0 \mid H_1)P(H_1) + P(H_1 \mid H_0)P(H_0)$$

$$\text{Assume : Correct classification is given zero cost} \Rightarrow C_{00} = C_{11} = 0$$

$$\text{Incorrect classification is given full cost} \Rightarrow C_{01} = C_{10} = 1$$

Statistical Feature Extraction and Selection are Key to Effective Target Classification

- **Assume we wish to classify targets into two classes** H_0 and H_1

- **Assume we are given a set of** $B \times 1$ **feature vectors** $\underline{X} = \left[x_1 \ x_2 \ , \cdots , \ x_B \right]^T$ **that have been extracted from the measurements**

- **We wish to minimize the number or features** B **that we use for several reasons:**
  - **- The curse of dimensionality**
  - **- To avoid over-fitting the data and reducing classification performance**
  - **- To avoid using features that are correlated enough that they do not contribute new information**

The Goal of Feature Selection:

Given: **A set of feature vectors containing** B **features**

Select: **A subset of** b **features** $(b \leq B)$ **that minimize class separation in feature space (minimize the distance in feature space between** H_0 and H_1 **)**

Statistical Feature Extraction and Selection are Key to Effective Target Classification

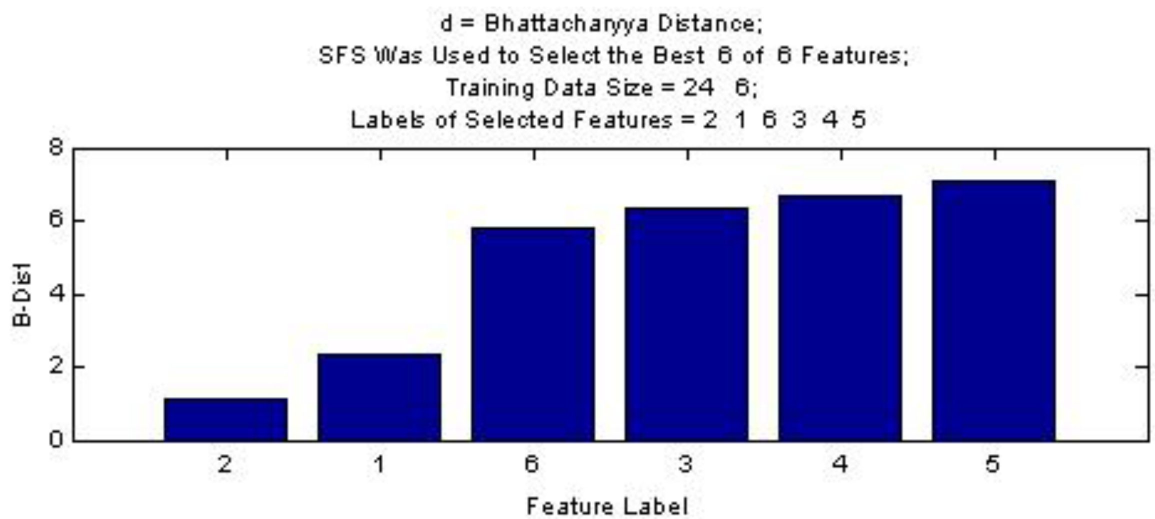
- Feature extraction/selection is the most important part of the target recognition process (Garbage in, Garbage Out)
- Most target recognition systems either use no feature selection or assume Gaussian distributed data
  - Suboptimal at best for non-Gaussian data (most real-world data!)
  - Wasteful of computational capacity
- Commonly-used feature selection algorithms use the Gaussian assumption because it is mathematically tractable and can be executed quickly
- The goal of this research is to create a practical feature selection algorithm for non-Gaussian data

Another Point of View

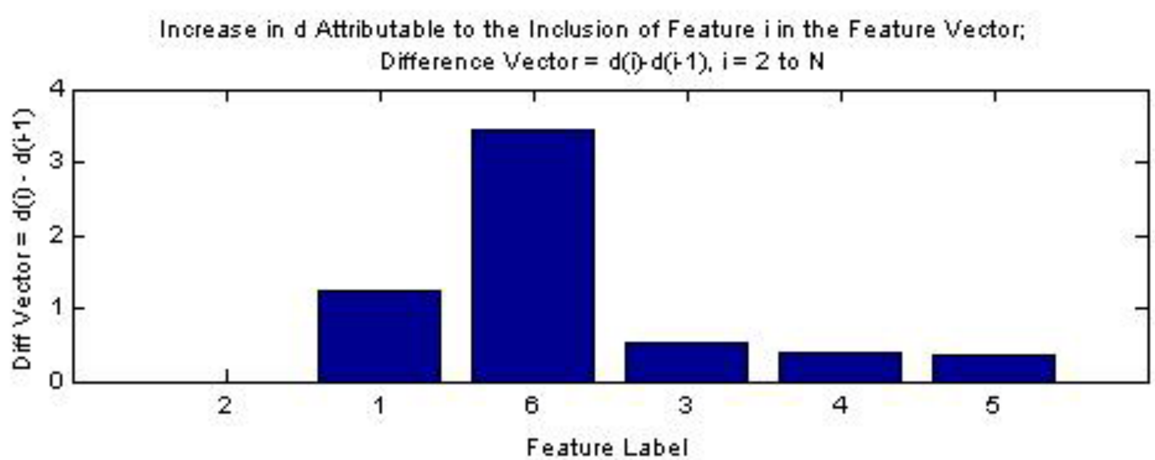
Photo From Lawrence A. Klein, Ph.D.

Sequential Forward Selection: "PAT Data"

Bhattacharyya Distance and a Small Data Subset

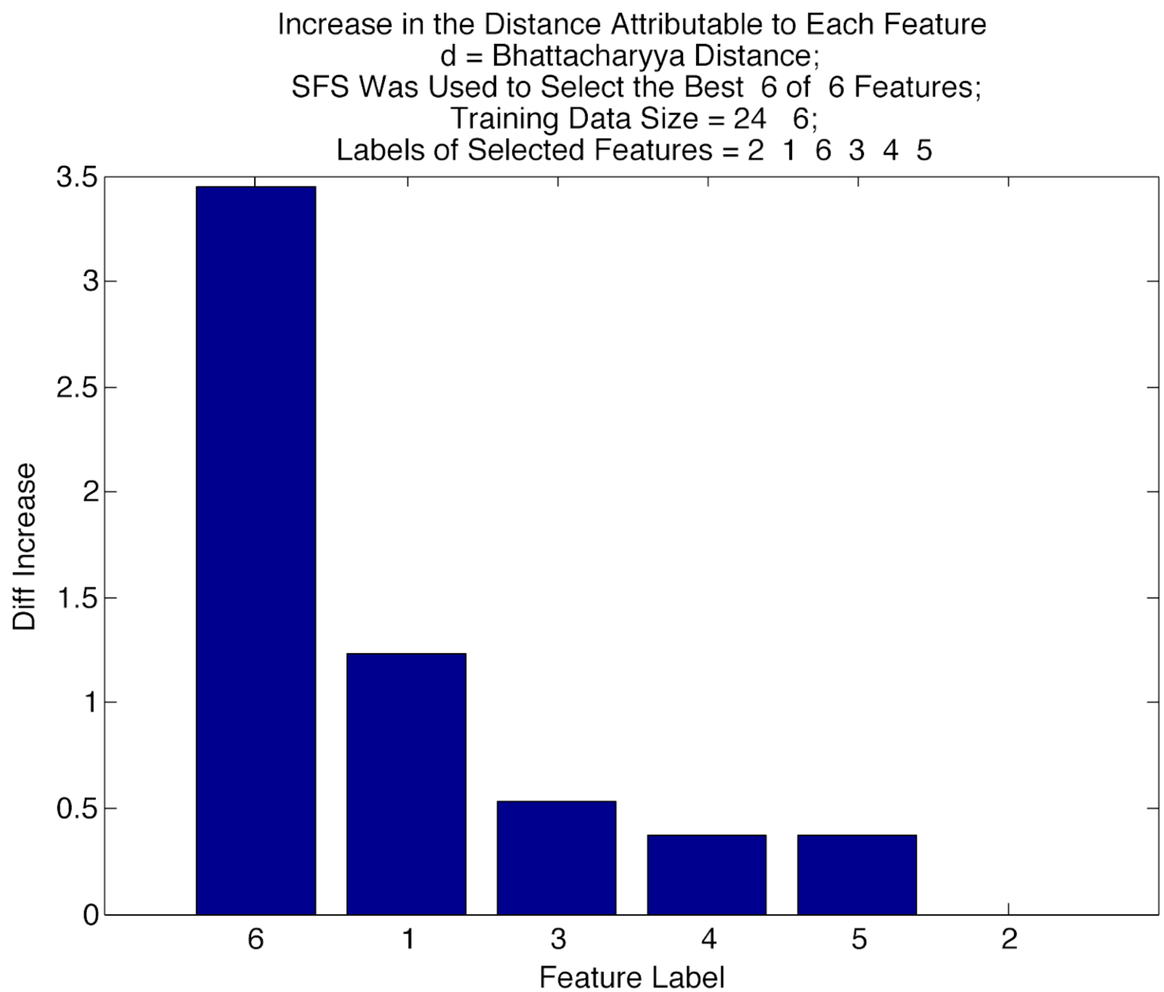


Distance vs. Feature Label



UNRANKED INCREASE **in the distance attributable to each feature**

Sequential Forward Selection: "PAT Data"
Bhattacharyya Distance and a Small Data Subset

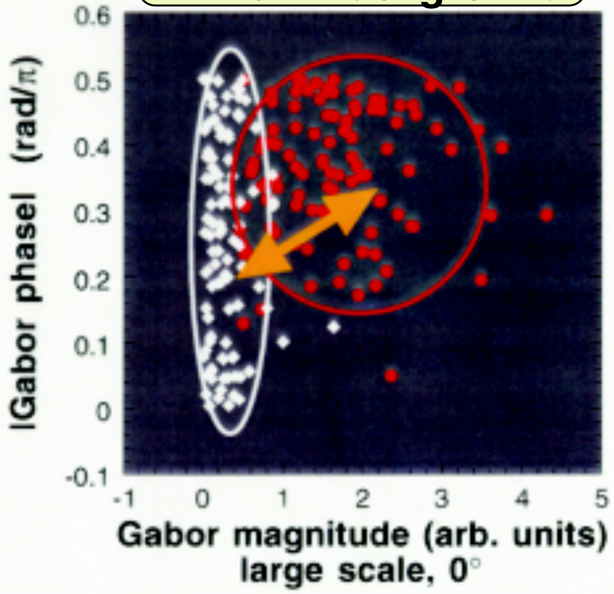


Here, we plot the *RANKED INCREASE* in the distance attributable to each feature

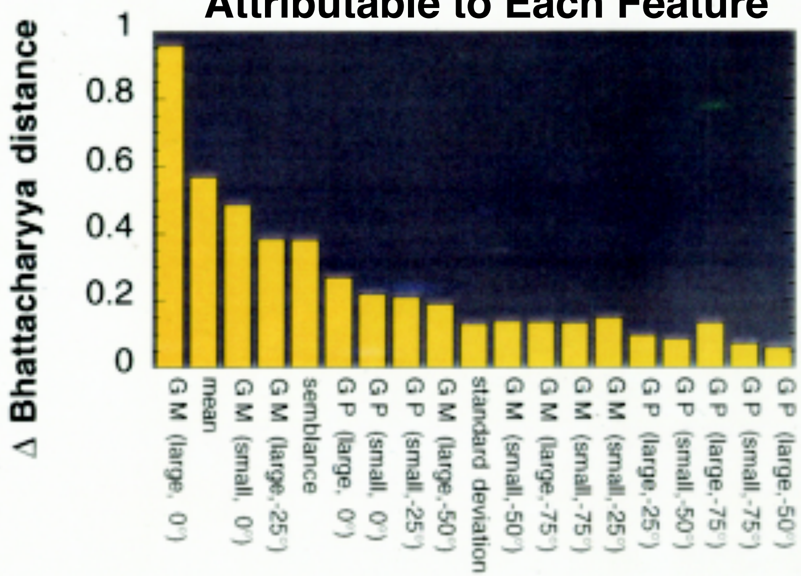
Feature Selection Example: *Automatic Event Picking for Seismic Oil Exploration (w/Shell Oil)*

Rank Order the Features According to the Change In the Bhattacharyya Distance, Using Sequential Feature Selection

Red = Events
White = Background



Increase in the Bhattacharyya Distance Attributable to Each Feature



distance between event and background cluster used

GM = magnitude of Gabor transform
GP = phase of Gabor transform

Lawrence Livermore National Laboratory

Feature Subset Selection for Gaussian Target Classes

Grace A. Clark

Commonly-Used Distance Measures Assume Gaussian-Distributed Target Classes

- **Define the Following Quantities for Two Multivariate Gaussian r.v.'s:**

$$\underline{\mu}_i = \text{Mean of the Data in Class } i, \qquad \Sigma_i = \text{The Covariance Matrix of Class } i$$

$$\underline{\mu}_j = \text{Mean of the Data in Class } j, \qquad \Sigma_j = \text{The Covariance Matrix of Class } j$$

- **The Mahalanobis Distance for Gaussian Data is:**

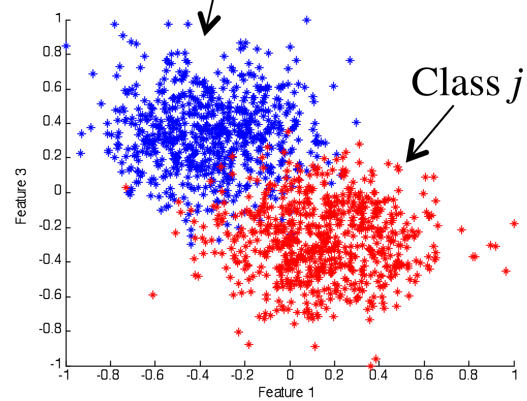
$$J_M(i,j) = (\underline{\mu}_i - \underline{\mu}_j)^T \left[\frac{\Sigma_i + \Sigma_j}{2}\right]^{-1} (\underline{\mu}_i - \underline{\mu}_j)$$

- **The Bhattacharyya Distance for Gaussian Data is:**

$$J_B(i,j) = \underbrace{\frac{1}{8}(\underline{\mu}_i - \underline{\mu}_j)^T \left[\frac{\Sigma_i + \Sigma_j}{2}\right]^{-1} (\underline{\mu}_i - \underline{\mu}_j)}_{\text{Mahalanobis Distance}} + \frac{1}{2}\ln\left[\frac{\left|\frac{\Sigma_i + \Sigma_j}{2}\right|}{\left|\Sigma_i\right|^{\frac{1}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}}\right]$$

Class i

Class j



Clusters of Data for the Two Classes in Feature Space

Feature Subset Selection Algorithms Vary in Complexity

- **Exhaustive Search:**

$$B = \text{Number of Available Features}$$

$$\text{b} = \text{ Number of Desired Features to Use}$$

$$\text{The Number of Possible Subset Combinations } = \binom{B}{b} = \frac{B!}{b!(B-b)!}$$

- Finds the globally-optimum feature subset
- *The curse of dimensionality dominates!*

- **Branch and Bound:**

- Finds the globally-optimum feature subset
- Saves computational complexity by not exploring every possible subset combination, when used with a monotonic class separation criterion.

The *Branch and Bound* Algorithm Rejects Suboptimal Subsets without Direct Evaluation

- **Yields the globally optimum solution when then the class separation criterion satisfies the *monotonicity condition:***

Let $J_i(x_1, x_2, \ldots, x_i) =$ The separation measure evaluated for all features $x_1, x_2, \ldots, x_i$ from the feature set.

$$J_1(x_1) \le J_2(x_1, x_2) \le \cdots \le J_b(x_1, x_2, \ldots, x_b)$$

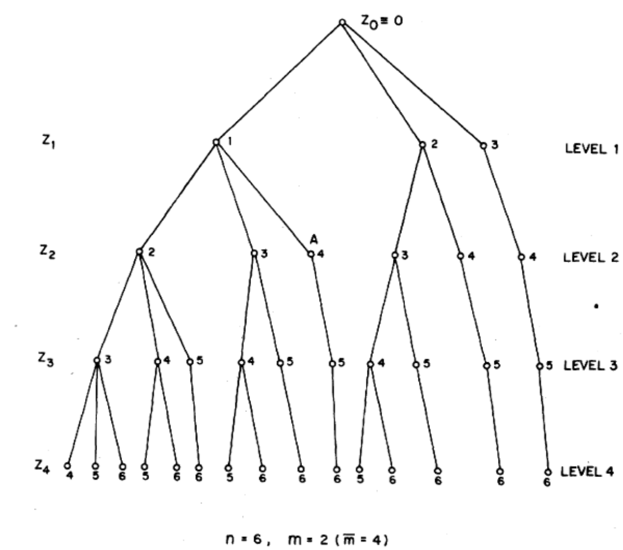
So, including more features should make the separation measure larger

The *Branch and Bound* Algorithm Rejects Suboptimal Subsets without Direct Evaluation

- Start with the full set of features
  - Define the initial "bound value" = the value of the separation measure at the bottom-right side of the decision tree.
- As we branch down each level of the decision tree, a feature is discarded.
- The separation measure is evaluated at each node and compared to the current bound level.

  If the a node higher in the tree provides a separation measure less than the bound, then the solutions stemming from that node do not require evaluation and are ignored.

  The current bound is updated according to various algorithms, depending on the variation of the B&B algorithm.



The Sequential Forward Selection (SFS) Algorithm Uses a Bottom-Up Search Strategy

- Start with a single feature
  - Add a feature to the current subset if the feature causes the separation measure to increase.
  - Remove a feature from the current subset if the feature causes the separation measure to decrease

    … and discard this feature from further consideration!

- SFS cannot guarantee optimality!
  - The best overall combination cannot necessarily contain the top-ranked available features, because some of those features may have been discarded!

- SFS runs very fast

- My experience over 20 years has shown that SFS generally performs well-enough for Gaussian data sets

- *Sequential Backward Selection* uses a similar "Top-Down" Strategy

Details of the Sequential Forward Selection Algorithm:

Table 1: The Sequential Forward Selection Algorithm

GIVEN: A set $\underline{X}$ containing B features from which it is desired to select a subset of b features ($b \leqslant B$).
Let $\underline{X}_k$ denote the selected feature set at algorithm iteration k.

1. Initialize the algorithm by setting the feature set $\underline{X}_0 = \{\}$ (the null set)

2. Specify b, the number of features desired in the final selected feature set.

3. kth Iteration ($k = 1, 2, 3, \ldots, b-1$)

   Rank order the elements a_i of the set $\underline{X} - \underline{X}_k$ so that

$$C(\underline{X}_k \cup \{a_1\}) \geqslant C(\underline{X}_k \cup \{a_2\}) \geqslant C(\underline{X}_k \cup \{a_{(B-k)}\}) \tag{12}$$

   Let $\underline{X}_{k+1} - \underline{X}_k \cup \{a1\}$

4. Repeat step 3 until $k + 1 - b$, then stop. The final feature set is $\underline{X}_b$.

Lawrence Livermore National Laboratory

Feature Subset Selection for non-Gaussian Target Classes

Grace A. Clark

It is Desired that the Distance Measure Satisfy the Four Properties of a Metric *(But Many Do Not)*

Let $d\left[f(x), g(x)\right]$ denote the distance between pdf's $f(x)$ and $g(x)$.
Then, the Four Properties of a Metric are:

(1) **Identity:** $d[f(x), g(x)] = 0$ if *f(x)=g(x)*. The distance between two identical objects should be zero.

(2) **Non-Negativity:** $d[f(x), g(x)] \geq 0$. To conform to traditional concepts of distance, the distance should be non-negative.

(3) **Symmetry:** $d[f(x), g(x)] = d[g(x), f(x)]$. The distance between *f(x)* and *g(x)* is the same as the distance from *g(x)* to *f(x)*.

(4) **Triangle Inequality:** $d[f(x), h(x)] \leq d[f(x), g(x)] + d[g(x), h(x)]$.

A distance should obey the property that the distance between *f(x)* and *g(x)* plus the distance between *g(x)* and *h(x)* should be less than or equal to the distance between *f(x)* and *h(x)*. This allows the distances among objects to be compared easily and reinforces the traditional concept of distance.

Information-Theoretic Distance Measures:
Divergence = Relative Entropy

Let $d(f,g)$ denote the distance between pdf's $f(x)$ and $g(x)$.

- **Kullback-Liebler (KL) Divergence:** $$d_{KL}(f,g) = \int_X g(x)\log\left(\frac{g(x)}{f(x)}\right)dx$$

  - Satisfies the Identity and Non-Negativity properties
  - *Is NOT symmetric and does not satisfy the Triangle Inequality*

- **Symmetric Kullback-Liebler (KLS) Divergence:**

$$d_{KLS}(f,g) = d_{KL}(f,g) + d_{KL}(g,f)$$

  - Satisfies the Identity, Non-Negativity and Symmetry properties
  - *Does not satisfy the Triangle Inequality*

- **Bhattacharyya Divergence:** $$d_B(f,g) = \int_X \sqrt{f(x)g(x)}dx$$

  - Satisfies the Identity, Non-Negativity and Symmetry properties
  - *Does not satisfy the Triangle Inequality*

The Hellinger Divergence
Satisfies all Four Properties of a Metric

- **The Squared Hellinger Divergence is:**

$$d_H^2(f,g) = \tfrac{1}{2} \int_X \left[\sqrt{f(x)} - \sqrt{g(x)}\right]^2 dx$$

- **The Hellinger Divergence is:**

$$d_H(f,g) = \tfrac{1}{\sqrt{2}} \sqrt{\int_X \left[\sqrt{f(x)} - \sqrt{g(x)}\right]^2 dx}$$

We Use the Hellinger Divergence because it satisfies the properties of a metric, it is robust, and it has the Monotonicity Property

Lawrence Livermore National Laboratory

pdf (Probability Density Function) Estimation

Grace A. Clark

A Multivariate Kernel Density Estimator Using a *Gaussian Kernel* is Commonly Used (e.g. in the Probabilistic Neural Network PNN)

We estimate the multivariate probability density function (pdf) of a random Vector $\underline{X}$ by summing kernel functions K(.) centered at the locations of the observations $\underline{X}_i$ (measurements)

$$\hat{f}(\underline{X}) = \frac{1}{(2\pi)^{\frac{p}{2}}\sigma^p}\frac{1}{M}\sum_{i=1}^{M}\exp\left[\frac{-(\underline{X}-\underline{X}_i)^T(\underline{X}-\underline{X}_i)}{2\sigma^2}\right]$$

$\underline{X}_i = p \times 1$ Measurement (training) data vector (i - th of M vectors)

$= \left[x_{i1}\ x_{i2} \cdots x_{ip}\right]^T$

i = Integer measurement (training) vector index over the range $[1, M]$

M = Integer Number of measurements $\underline{X}_i$ available for training

p = Integer dimension of the measurement space

$\underline{X}$ = $p \times 1$ Grid vector point at which we wish to evaluate the estimate of the pdf

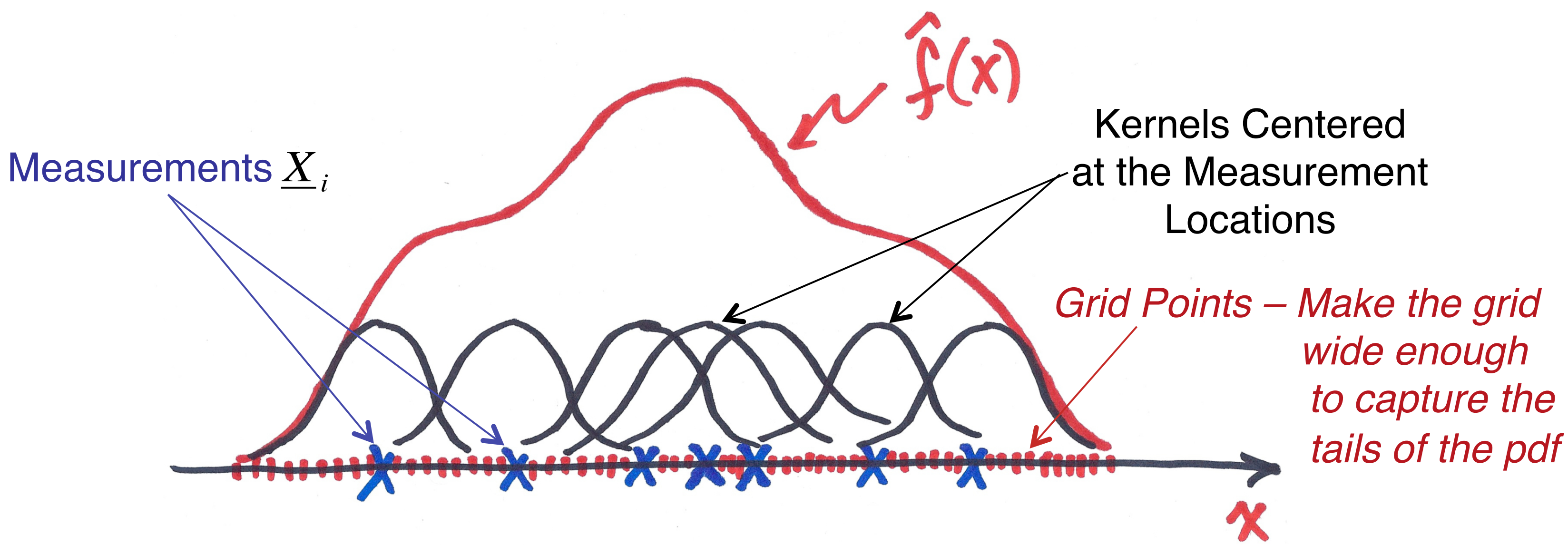
σ = Scalar real - valued smoothing parameter or window width

Example pdf Estimation for the Univariate Case (p=1): We Must Build a "Grid" of Samples at Which We Wish to Estimate the pdf

$$\hat{f}(\underline{X}) = \frac{1}{(2\pi)^{\frac{p}{2}}\sigma^{p}} \frac{1}{M} \sum_{i=1}^{M} \exp\left[\frac{-(\underline{X}-\underline{X}_i)^T(\underline{X}-\underline{X}_i)}{2\sigma^2} \right]$$

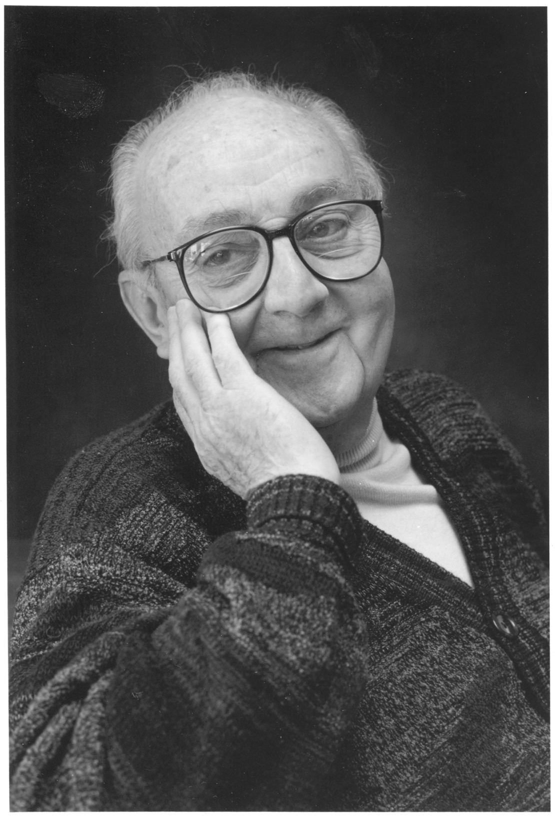
Draw an "X" at the location along the real line of each of the measured data samples that are available for training.

Draw a Red hash mark "I" along the real line at the locations of the desired "grid points" at which we wish to estimate the pdf values.



George E. P. Box (*10/18/1919 -   *)

Professor Emeritus of Statistics at the University of Wisconsin, and a pioneer in the areas of quality control, time series analysis, design of experiments and Bayesian inference.

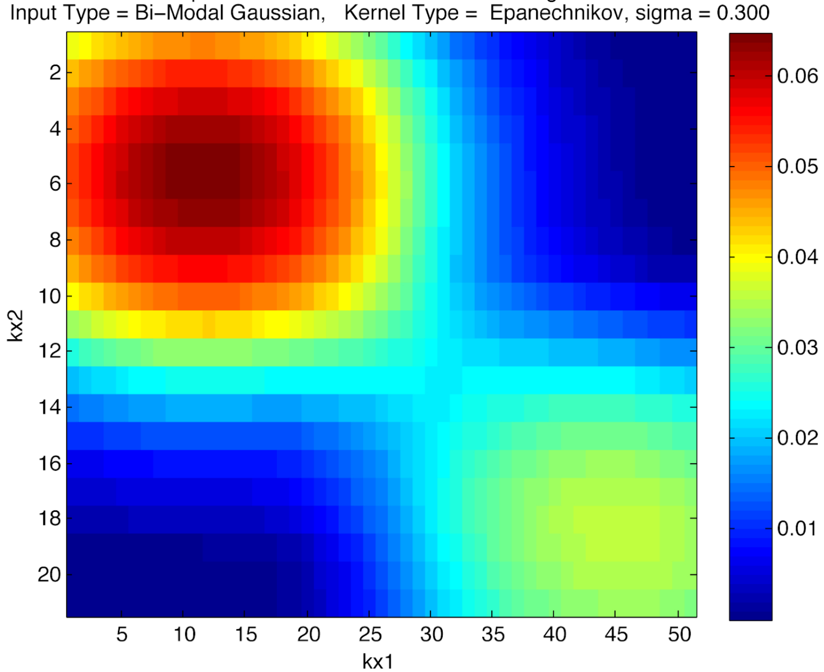


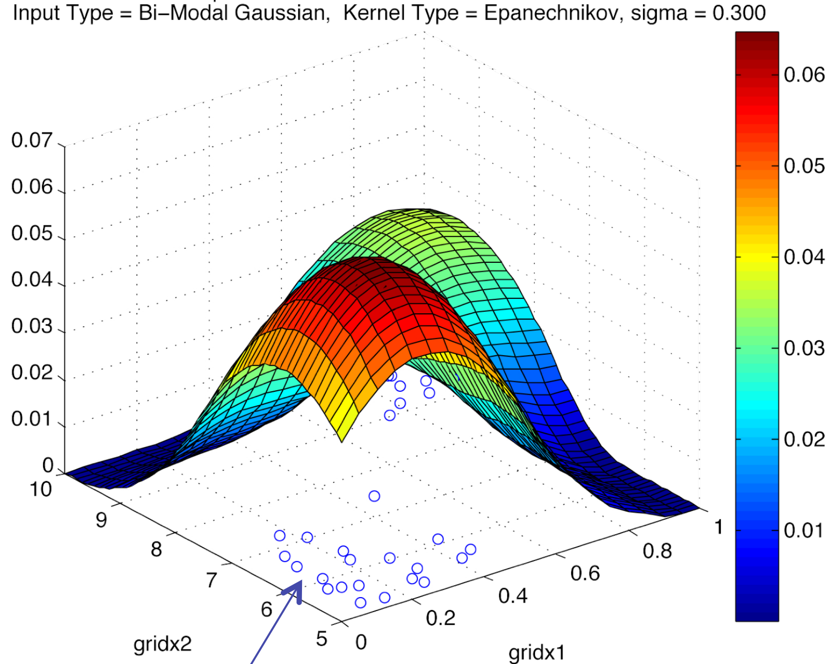
"Essentially, all models are wrong, but some are useful."

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."

Example: *Kernel Density Estimate* for a ***Bivariate, Bi-Modal Gaussian r.v.*** Using an *Epanechnikov Kernel*

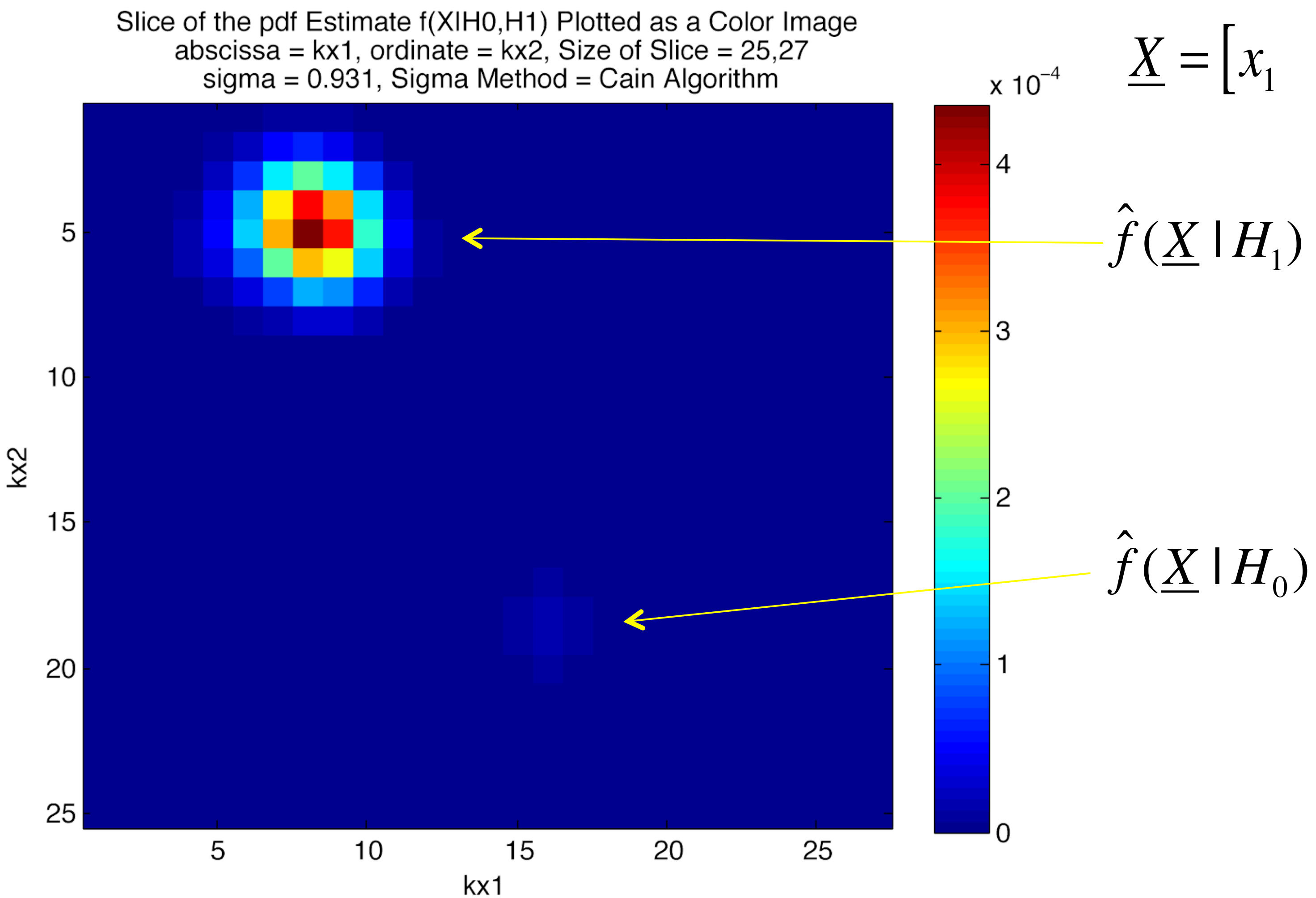
$$\underline{X} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \qquad \hat{f}(\underline{X})$$





Measurement Vectors $\underline{X}_i$ *Used to Estimate the pdf*

Example: A *Slice* of Feature Space for the 3D Kernel Density Estimates of $\hat{f}(\underline{X} \mid H_1)$ and $\hat{f}(\underline{X} \mid H_0)$



$$\underline{X} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$$

Lawrence Livermore National Laboratory

Our New Feature Selection Algorithm for Non-Gaussian Data

Grace A. Clark

Our New Feature Selection Algorithm is Tested Using a *Bayes Classifier / Probabilistic Neural Network*

- **The New Feature Selection Algorithm for Non-Gaussian Data Uses:**
  - ***Distance Measure:***
    - **- Hellinger Divergence**
  - ***Subset Selection Algorithms:***
    - **- Sequential Forward Selection (SFS)**
    - **- Branch and Bound**
  - ***pdf Estimator:***
    - **- Kernel Density Estimator with a Gaussian Kernel**

- **We Compare Results with a Classical FS Algorithm for Gaussian Data:**
  - ***Distance Measure:***
    - **- Bhattacharyya and Mahalanobis Distances**
  - ***Subset Selection Algorithms:***
    - **- Sequential Forward Selection (SFS)**
    - **- Branch and Bound**

Lawrence Livermore National Laboratory

Selected Experimental Results

Grace A. Clark

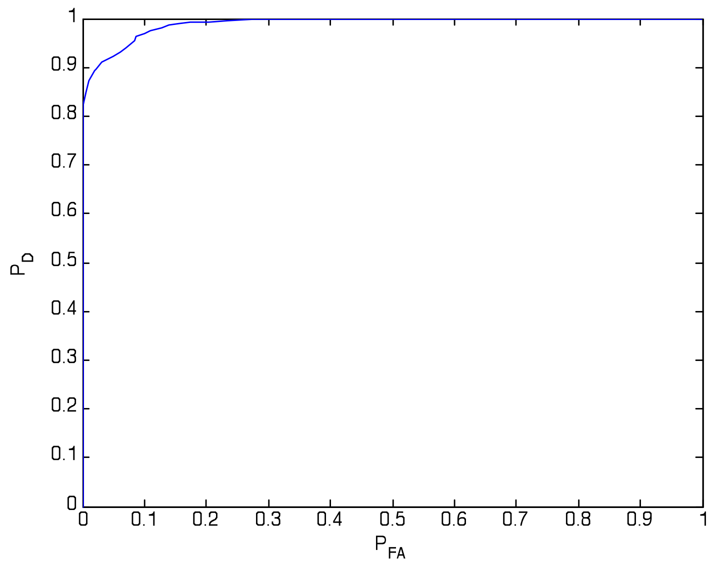
Experiment: Both Target Classes are Gaussian

Branch and Bound, Training Set = 700 FV's/Class, Test Set = 500 FV's/Class

- Select the best 2 of 8 features: Test Set = 700 FV's/ Class

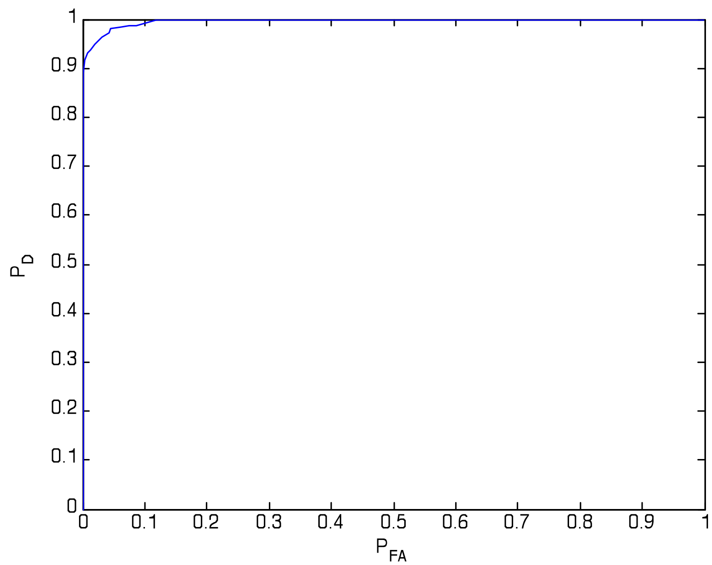
Hellinger → P(CC) = 94%

[3 5] = The subset chosen



Bhattacharyya →P(CC) = 96.9%

[1 3] = The subset chosen



Mahalanobis → P(CC) = 96.9% Same subset chosen as the Bhattacharyya

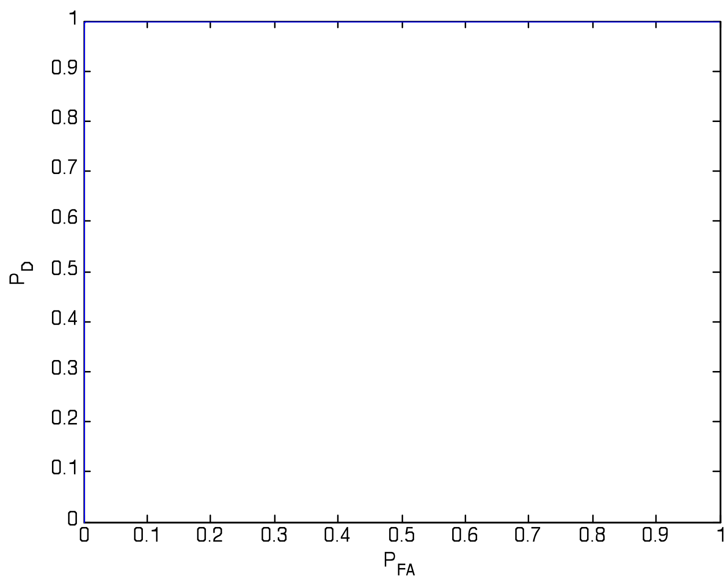
The "non-Gaussian" algorithm did not do as well as the Gaussian algorithm For truly Gaussian data. This is not surprising.

Experiment: One Gaussian and one non-Gaussian Target Class
Branch and Bound, Training Set = 700 FV's/Class, Test Set = 500 FV's/Class

- Select the best 2 of 8 features

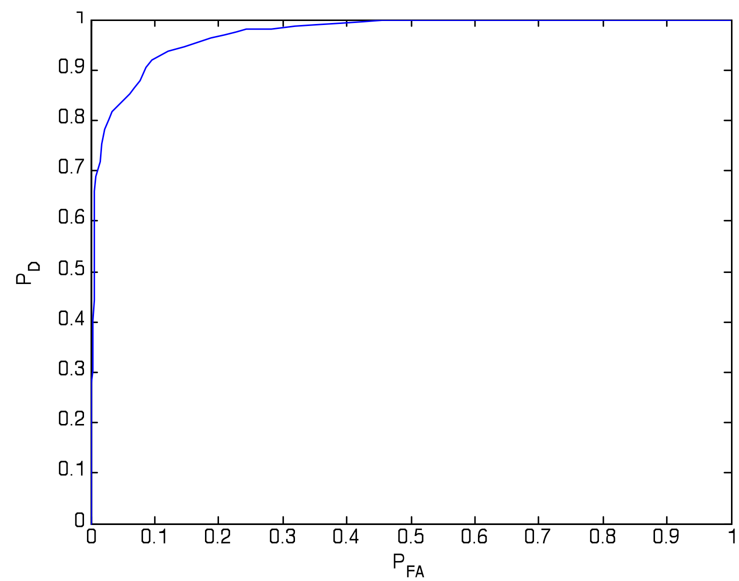
Hellinger→P(CC) = 99.0%

[3 8] = The subset chosen



Bhattacharyya→P(CC) = 91.40%

[1 3] = The subset chosen



Mahalanobis→ 91.40% Same subset of the Bhattacharyya

The "non-Gaussian" algorithm performed better than the "Gaussian" algorithm for non-Gaussian data (as expected).

Experiment: Both Target Classes non-Gaussian

Branch and Bound, Training Set = 470 FV's/Class, Test Set = 320 FV's/Class

- Select the best 2 of 8 features

Hellinger→ P(CC) = 93.6%

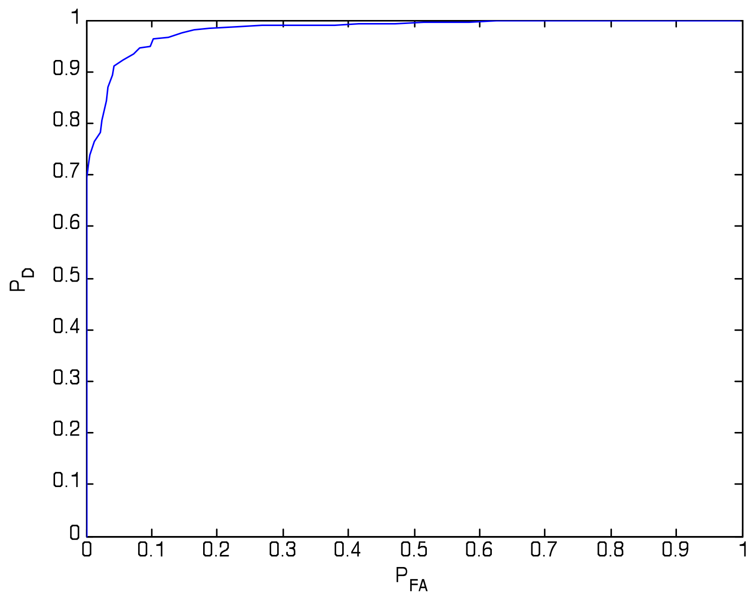
[3   5] = Chosen feature subset

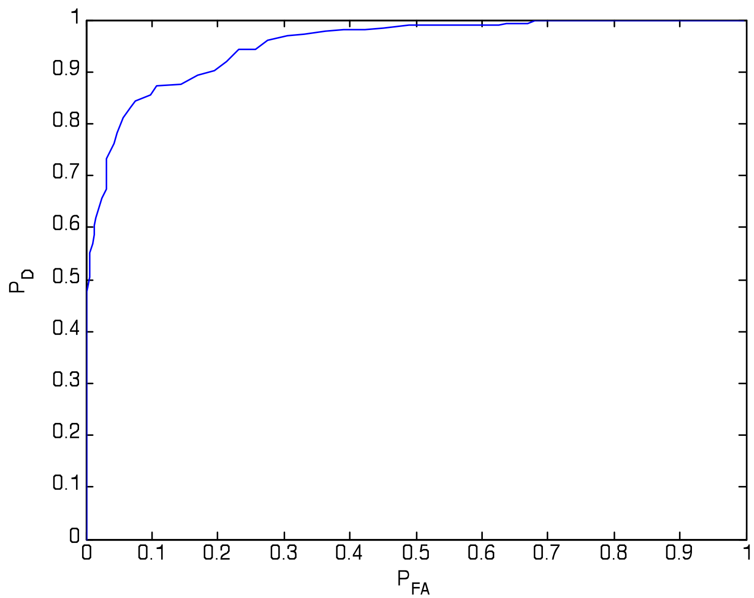
Bhattacharyya→ P(CC) = 88.6%

[5   7] = Chosen feature subset

Mahalanobis→ P(CC) = 58.9%

[3   4] = Chosen feature subset





The "non-Gaussian" algorithm performed much better than the "Gaussian" algorithm for non-Gaussian data (as expected).

We Use a Well-Known Real-World Benchmark Data Set:
“Classic Fisher Iris Data” *(Approx. Non-Gaussian)*

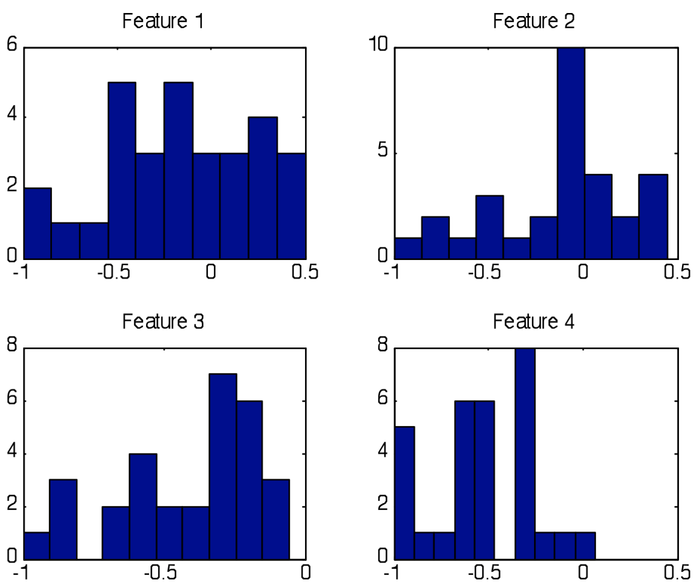
Number of Features = 4

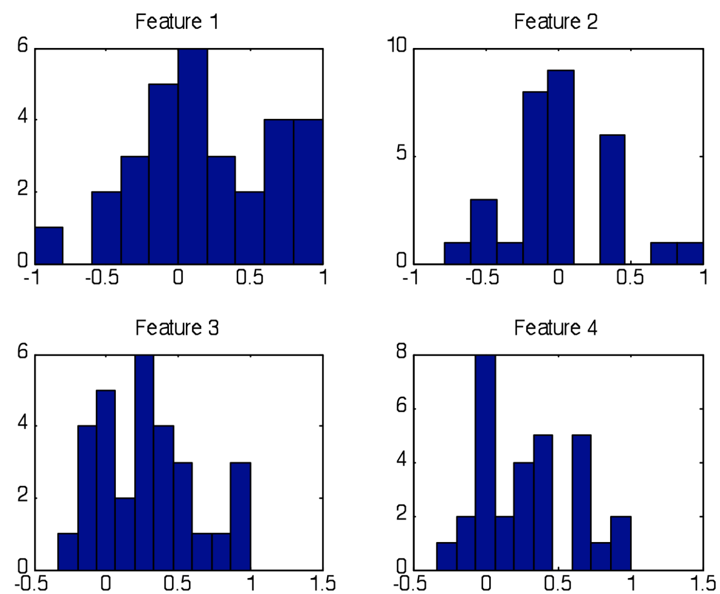
Hypothesis H_0

"Versacolor"

Hypothesis H_1

"Virginica"





We Use a Well-Known Real-World Benchmark Data Set: *"Classic Fisher Iris Data"* *(Approx. Non-Gaussian)*

- **The four features correspond to**
  - **- Sepal length**
  - **- Sepal width**
  - **- Petal length**
  - **- Petal width**

- **The data set contains 50 feature vectors per class (Two classes)**
- ***Training Set:***
  60% of the available 50 vectors per class
  ➔ 30 vectors per class
- ***Test Set:***
  40% of the available 50 vectors per class
  ➔ 20 vectors per class
- **Feature Subset Selection Algorithm Used:** ***Branch and Bound***

"Classic Fisher Iris Data"

• Select the best 3 of 4 features

Hellinger→P(CC) = 97.50%

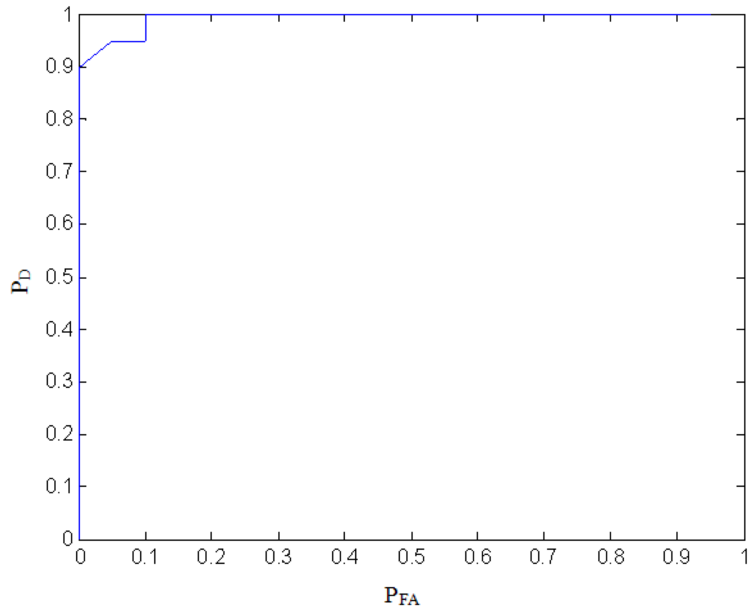
[2 3 4] = Chosen feature subset

Bhattacharyya→97.50%

[1 3 4] = Chosen feature subset

Mahalanobis→95.00%

[1 2 3] = Chosen feature subset



Feature Subset Selection Algorithm Used:

Branch and Bound

The "non-Gaussian" algorithm performed best. The "Gaussian" algorithm did well => The data were nearly Gaussian

Conclusions

- When the data are non-Gaussian, the new algorithm far outperforms the “Gaussian” algorithm
  - When the data are Gaussian ➔ use the classic algorithms

- **Future Research**
  - Increase Dimensionality
    - -- Above about 10 or 12 features, the computational complexity becomes very burdensome, due to density estimator
    - -- Explore other density estimation algorithms
  - Increase speed and efficiency
    - -- Algorithm optimization
    - -- Parallel processing, etc.
  - Test the algorithm with more and varied datasets

The World of Acoustics Before Signal Processing